REMARKS

This responds to the Office Action mailed on July 27, 2007.

Claims 7, 22, 24 and 29-32 are amended, claim 12 is canceled, and no claims are added; as a result, claims 7, 8, 14, 15, 22, 24, 25 and 29-32 are now pending in this application.

Claim Objections

Claims 7, 8, 12, 14 and 15 were objected to for informalities, namely, in claim 7, the term -- the cache -- should be corrected to -- the local cache--.

Claim 7 has been so amended to overcome this objection.

§112 Rejection of the Claims

Claims 24 and 25 were rejected under 35 U.S.C. § 112, second paragraph, for indefiniteness.

Claims 24 and 25 have been amended to overcome this objection.

§103 Rejection of the Claims

Claims 7, 8, 12, 14, 22, 24, 25 and 29-32 were rejected under 35 U.S.C. § 103(a) being unpatentable over Akkary (U.S. Patent Application Publication No. 2003/0196035; hereinafter "Akkary") in view of Hughes et al. (U.S. Patent No. 6,393,536: hereinafter "Hughes") in further view of Henry et al. (U.S. Patent Application Publication No. 2003/0018875: hereinafter "Henry") in further view of Merchant et al. (U.S. Patent No. 6,385,715: hereinafter "Merchant") in further view of Hennessy et al. (Computer Organization and Design: The Hardware / Software Interface: hereinafter "Hennessy").

Applicant's invention as claimed in claims 7, 22 and 29 pertains to processing memory requests in a scalar processor using an Initial Request Queue (IRQ) and a Force Order Queue (FOQ). Specifically, Applicant teaches, and claims in claims 7, 22 and 29, determining not only data match ('hit' or 'miss') in a local cache but also a partial address match in the FOQ for each memory request, servicing the memory request using a higher level cache if the memory request matches any FOQ entry, and allocating a new cache line in the local cache if the memory request misses in the local cache and in the FOQ simultaneously. Claims 7, 22 and 29 have been amended to more clearly define Applicant's claimed invention.

By routing the memory requests and allocating new cache lines selectively, Applicant's claimed invention reduces memory latency and simplifies cache line allocation when a cache miss occurs. As recognized by Applicant, at p. 1, line 20 through p. 2, line 4, this enhances overall performance in a large scale system because a cache miss occurs more often as an increased number of processors access distributed shared memory in the large scale system. To accomplish this, Applicant teaches and claims, for example, in amended claim 1:

> "...wherein the FOQ stores a memory request that is pending to the higher level cache;
>
> if the portion of an address associated with the memory request does not match the one or more partial addresses in the FOQ and, at the same time, the memory request hits in the local cache, servicing the memory request immediately using data in the local cache;
>
> if the portion of an address associated with the memory request does not match the one or more partial addresses in the FOQ and, at the same time, the memory request misses in the local cache, adding the memory request to the FOQ, allocating a cache line in the local cache corresponding to the local cache miss and servicing the memory request using data received from the higher level cache; and
>
> if the portion of an address associated with the memory request matches the one or more partial addresses in the FOQ, preventing the memory request from being satisfied in the local cache, wherein preventing includes adding the memory request to the FOQ and servicing the memory request using data received from the higher level cache."

Claims 22 and 29 also have corresponding limitations.

Akkary describes using a store buffer (Fig. 3, block 260) in combination with a data cache (Fig. 3, block 270) in a prioritized content addressable memory. See Abstract and Fig. 3.

Merchant describes a method and apparatus for replaying memory requests in a processor having two levels of local caches (i.e., L0 and L1 caches).

Hughes describes a load/store unit including two buffers, LS1 and LS2, for storing memory operations and two control logics attached to the buffers.

Henry describes virtual address translation (Fig. 2) which compares physical page index (204) of a given instruction.

Hennessy describes a mechanism to determine which block should be replaced on a cache miss. See page 606.

None of the cited references, however, alone or in combination, teach or using the FOQ that stores memory requests pending to the higher level cache, checking both the local cache and the FOQ miss/hit states simultaneously to determine whether to add the memory requests to the

FOQ and to determine whether to allocate a new cache line in the local cache upon a cache miss as taught by Applicant and claimed in claims 7, 22 and 29.

In the Office Action, at p. 5, lines 7-8, the Examiner states that the use of the store buffer ([0032], block 260) shows Applicant's limitation of "determining whether an address associated with the memory request matches an address in a Forced Order Queue (FOQ)."

This statement, while partially true, fails to recognize that Akkary's store buffer is different from Applicant's Forced Order Queue (FOQ). Under Akkary's approach, for store instructions, the store buffer (Fig. 3, block 260) holds data temporarily before an address of a location in the data cache where the data is to be stored is calculated and the data is transferred to the data cache (block 270). See [0030], lines 7-19 (with an emphasis on the lines 18-19, "the data that will be stored in the data cache (270) is stored in the entry [of the store buffer]"). For load instructions, the store buffer transfers the data requested by load instructions to its associated register file if the load address matches an address in the store buffer ("hits" the store buffer). See [0032], lines 5-10. In either case, therefore, the store buffer stores data (and its associated address) that is needed to service memory requests and provides the data to its associated local data cache (if the memory requests are store instructions) or to the Register File (block 240) (if the memory requests are load instructions).

In contrast, under Applicant's approach, the FOQ stores memory requests that are pending to a higher level cache (e.g., Ecache) upon a cache miss. Also, as noted in the Specification, p. 17, lines 3-7 and 17-20 and discussed above, the memory requests stored in the FOQ wait for its missed memory line (i.e., its required data) to be available in the higher level cache (e.g., Ecache) and then are serviced by the higher level cache. The requested memory line (i.e., data) is then sent from the higher level cache to the local cache for execution of memory requests subsequent to the cache miss. These differences are more clearly defined in amended claims 7, 22 and 29.

The Examiner also states that Akkary's "the selector (280) chooses the entry in the store buffer (260) if the load address hits the store buffer, else the data cache (270) fulfills the memory request" further teaches Applicant's limitation of "if a portion of an address associated with the memory request matches one or more partial addresses in the FOQ, preventing the memory request from being satisfied in the local cache." Office Action, p. 5, line 19 through p. 6, line 3.

Applicant respectfully submits that this statement fails to recognize the difference between Applicant's FOQ and Akkary's store buffer as discussed above. They serve different functions. Although data required for execution of a memory request is transferred from the store buffer (260) to the local data cache (270) or to the register file (240) as discussed above, Akkary does not use the store buffer to monitor cache misses and to allocate space in the local cache as a function of memory requests that miss in the local cache.

In contrast, Applicant teaches, at p. 17, lines 3-4 and 12-19, and claims in claim 7, adding memory requests to the FOQ not only upon a cache miss in the local cache but also upon a cache hit in the local cache if there is a partial address match in the FOQ. Applicant respectfully submits this difference is underscored by the amended limitation of "if the portion of an address associated with the memory request **matches the one or more partial addresses in the FOQ, ... adding the memory request to the FOQ** and servicing the memory request using data received from the higher level cache." Memory requests are not added to the FOQ, however, upon a hit in the local cache if the memory requests do not match any partial addresses in the FOQ. Applicant, therefore, uses both the local cache and the FOQ hit/miss conditions to determine whether to add the memory requests to the FOQ for their execution. Applicant is unable to find such a teaching in any of the cited references. Claims 7, 22 and 29 have been amended to emphasize this difference.

Finally, the Examiner states that Hennessy's cache line allocation mechanism and Merchant's replay queue (Fig. 1, block 170 and col. 8, lines 13-22) in further combination with Akkary's store buffer teaches Applicant's limitation of "if the memory request misses in the cache and, at the same time, the address does not match one or more partial addresses in the FOQ, adding the memory request to the FOQ and allocating a cache line in the local cache corresponding to the local cache miss." Office Action, p. 6, line 4 through p. 7, line 6.

Hennessy discusses allocating a new cache line. None of the references, however, teach or suggest checking both the local cache and the FOQ hit/miss conditions simultaneously to determine whether to allocate a new cache line upon a cache miss by a memory request.

Under Merchant's approach, the replay system uses the Replay Queue (Fig. 1, block 170) to replay requests which have not executed properly, for example, because of a cache miss. When a cache miss occurs, the request goes through either the replay queue (170) or the replay

loop (Fig. 1, block 156). If the request misses both two levels of local caches, L0 and L1, (i.e., a long latency instruction) and, therefore, needs an external bus request, the request will be loaded into the replay queue. If the request misses the L0 cache and hits the L1 cache (i.e., not a long latency instruction), the request will not be loaded into the replay queue. Instead, it will go through the replay loop for execution. See col. 7, lines 1-13, lines 24-42; col. 8, lines 13-13-39; col. 8, line 62 through col. 9, line 8. Merchant, therefore, checks a hit or miss condition in the two local caches (L0 and L1) to determine whether to add the memory requests to the replay queue (170) and possibly to determine whether to allocate new cache lines for cache misses in L0 and L1. Merchant does not, however, teach or suggest checking a partial address hit or miss in the reply queue (170) to determine whether a new cache line needs to be allocated for a memory request upon its cache miss. Under Merchant's approach, therefore, it is possible that a new cache line is repeatedly allocated for the same memory line if the memory line was requested from other sources (e.g., a higher level cache or memory) upon a first memory request's cache miss but not received to the local cache yet.

In contrast, as noted in the Specification at p. 17, lines 3-7 and 17-20, under Applicant's approach, a new cache line is not allocated in a local cache even when a memory request misses the local cache if the memory request matches a partial address in any entries in the FOQ. Instead, the memory request are routed to the FOQ and passed on to the Ecache, waiting for the missed data to be available in the Ecache for execution. This is important to enhance performance of a processor in a large scale system with an increased number of processors accessing shared distributed memory as discussed above. Applicant is unable to find such a teaching in any of the cited references.

The Examiner notes that Akkary fails to teach a procedure for fulfilling a memory request when the memory request misses in both the local cache and the FOQ but states that Merchant describes the use of a replay queue. Merchant's replay queue is discussed above. Applicant respectfully submits that the combination of a store buffer to store intermediate data in Akkary and a replay queue to store cache-missed requests in Merchant is inoperable since there is no indication in any of the references that would suggest when the store buffer should be used instead of the replay queue, and vice versa.

For these reasons, none of the references cited by the Examiner, alone or in combination, teach or suggest using the FOQ that stores memory requests pending to the higher level cache, checking both the local cache and the FOQ miss/hit states simultaneously to determine whether to add the memory requests to the FOQ and to determine whether to allocate a new cache line in the local cache upon a cache miss as taught by Applicant and claimed in claims 7, 22 and 29. Reconsideration is respectfully requested.

With regard to claims 30, claim 30 is patentable as being dependent on a patentable base claim. In addition, none of the cited references, alone or in combination, teach or suggest using a FOQ index array associated with the FOQ as taught by Applicant and claimed in claim 30.

With regard to claims 31 and 32, claims 31 and 32 are patentable as being dependent on a patentable base claim. In addition, none of the cited references, alone or in combination, teach or suggest dividing the FOQ into a first and second queue and having each queue monitors memory requests to a higher level queue and memory requests that are serviced by the higher level cache but not written to the local cache, respectively as taught by Applicant and claimed in claims 31 and 32.

With regard to claims 8, 12, 14, 24 and 25, claims 8, 12, 14, 15, 24 and 25 are patentable as being dependent on a patentable base claim.

Claim 15 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Akkary in view of Hughes et al., in further view of Henry et al., in further view of Merchant et al., in further view of Hennessey et al., as applied to claims 7, 8, 12, 14, 22, 24, 25 and 29-32 above, in further view of Yamahata (US 5,247,639).

Claim 15 is patentable as being dependent on a patentable base claim.

RESERVATION OF RIGHTS

In the interest of clarity and brevity, Applicant may not have addressed every assertion made in the Office Action. Applicant's silence regarding any such assertion does not constitute any admission or acquiescence. Applicant reserves all rights not exercised in connection with this response, such as the right to challenge or rebut any tacit or explicit characterization of any reference or of any of the present claims, the right to challenge or rebut any asserted factual or legal basis of any of the rejections, the right to swear behind any cited reference such as provided under 37 C.F.R. § 1.131 or otherwise, or the right to assert co-ownership of any cited reference. Applicant does not admit that any of the cited references or any other references of record is relevant to the present claims, or that they constitute prior art. To the extent that any rejection or assertion is based upon the Examiner's personal knowledge, rather than any objective evidence of record as manifested by a cited prior art reference, Applicant timely objects to such reliance on Official Notice, and reserves all rights to request that the Examiner provide a reference or affidavit in support of such assertion, as required by MPEP § 2144.03. Applicant reserves all rights to pursue any cancelled claims in a subsequent patent application claiming the benefit of priority of the present patent application, and to request rejoinder of any withdrawn claim, as required by MPEP § 821.04.

CONCLUSION

Applicant respectfully submits that the claims are in condition for allowance, and notification to that effect is earnestly requested. The Examiner is invited to telephone Applicant's attorney at (612) 373-6909 to facilitate prosecution of this application.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 19-0743.

Respectfully submitted,

SCHWEGMAN, LUNDBERG & WOESSNER, P.A.
P.O. Box 2938
Minneapolis, MN 55402
(612) 373-6909

Date January 23, 2008 By _____
Thomas F. Brennan
Reg. No. 35,075

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Mail Stop Amendment, Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on this ~~23~~ 24 day of January 2008.

CANDIS BUENDING _____

Name Signature